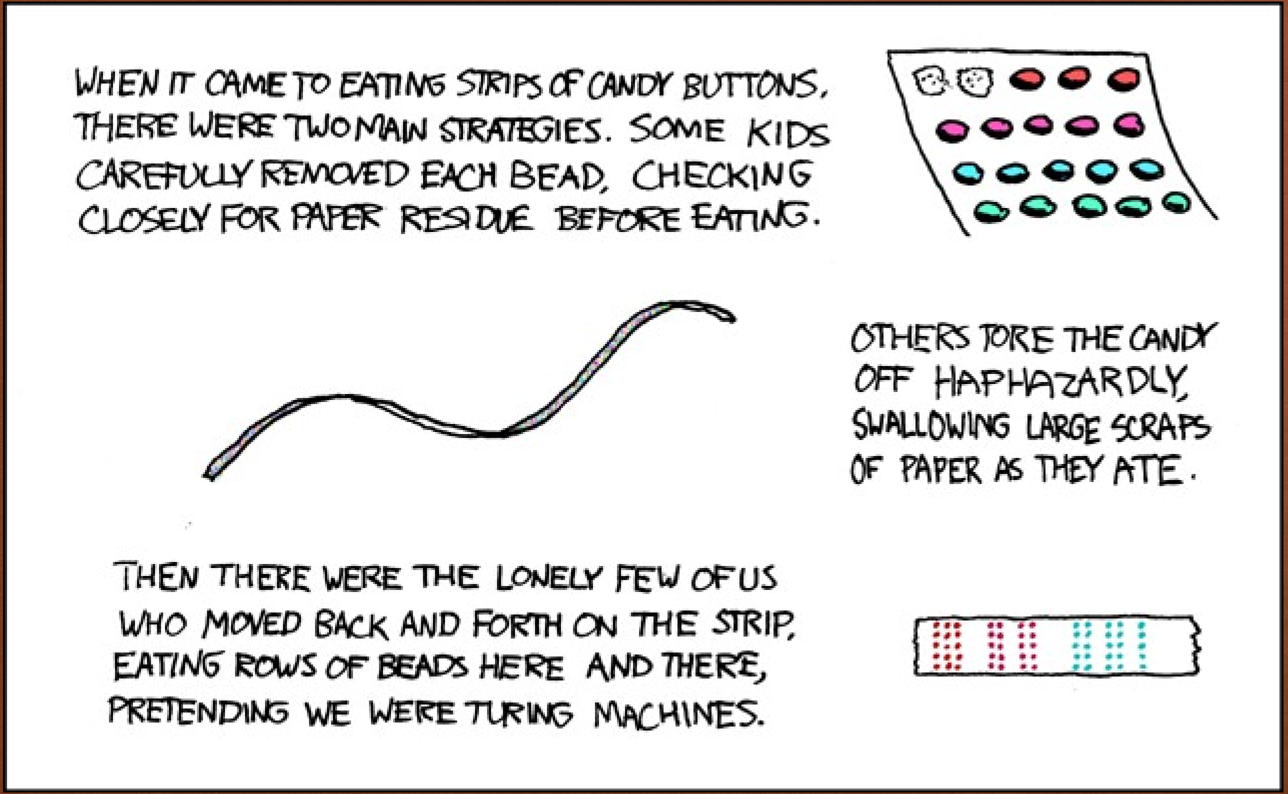
TURING MACHINES
Chapter 12


WHEN IT CAME TO EATING STRIPS OF CANDY BUTTONS, THERE WERE TWO MAIN STRATEGIES. SOME KIDS CAREFULLY REMOVED EACH BEAD, CHECKING CLOSELY FOR PAPER RESIDUE BEFORE EATING.
OTHERS TORE THE CANDY OFF HAPHAZARDLY, SWALLOWING LARGE SCRAPS OF PAPER AS THEY ATE.
THEN THERE WERE THE LONELY FEW OF US WHO MOVED BACK AND FORTH ON THE STRIP, EATING ROWS OF BEADS HERE AND THERE, PRETENDING WE WERE TURING MACHINES.

https://xkcd.com/205/

TURING MACHINE EXAMPLES

Parity bit problem

Given a string of 0s and 1s,

Count whether the number of 1s is even or odd and add a parity bit accordingly

For **odd parity**, the number of 1’s in a string must be odd.

For **even parity**, the number of 1’s in a string must be even.

Let’s say we want to write a Turing machine for **odd parity**.

If the number of 1s in a string is even, add a 1 to the end of the string.

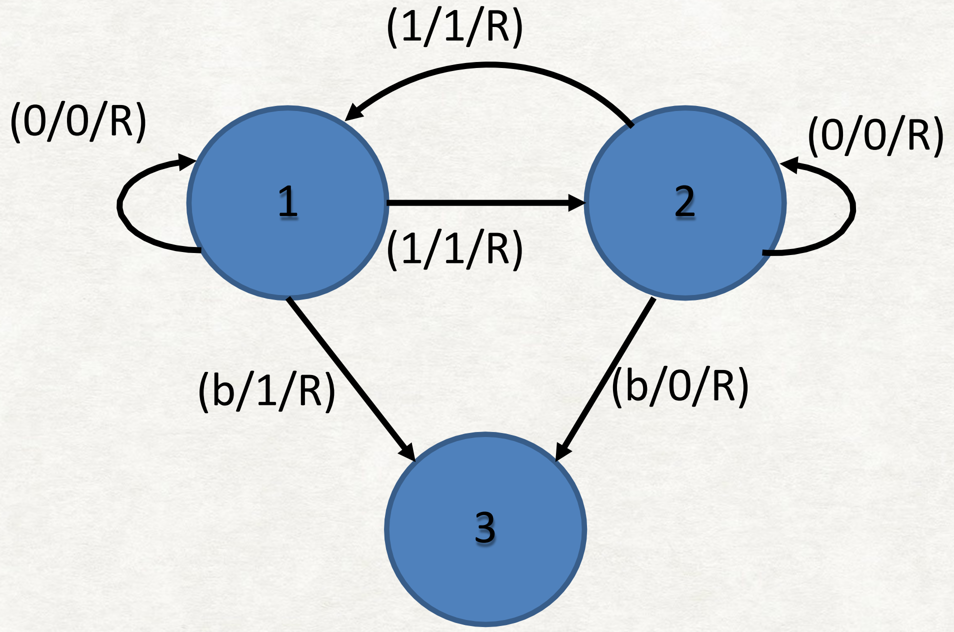
e.g. 1 1 0 0 1 0 1 0 1

If the number of 1s in a string is odd, add a 0 to the end of the string

e.g. 1 0 1 1 0 1 0 1 0

TURING MACHINE EXAMPLES : PARITY BIT MACHINE

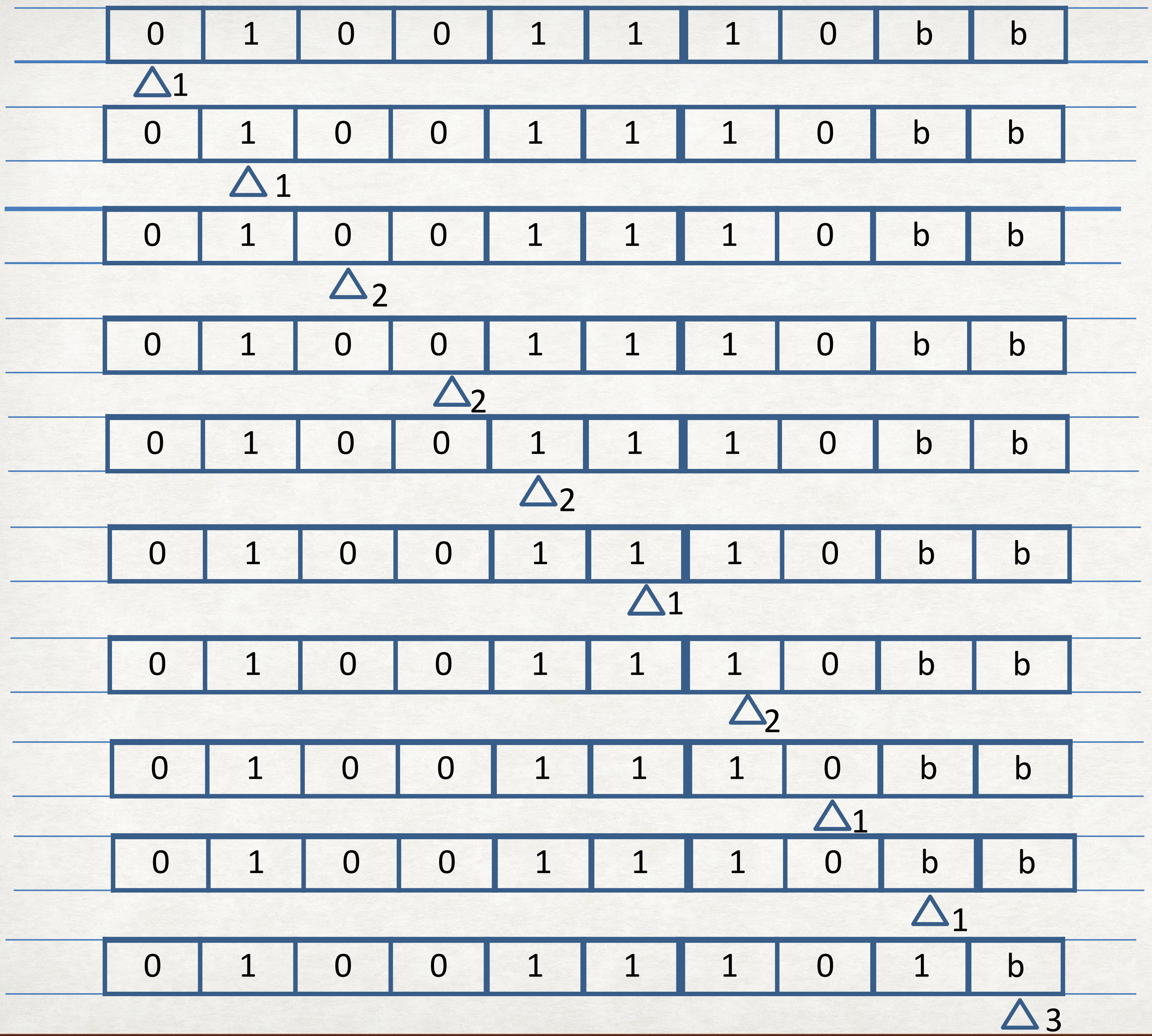
- State 1: number of 1's seen so far is even
- State 2: number of 1's seen so far is odd
- State 3: halting state



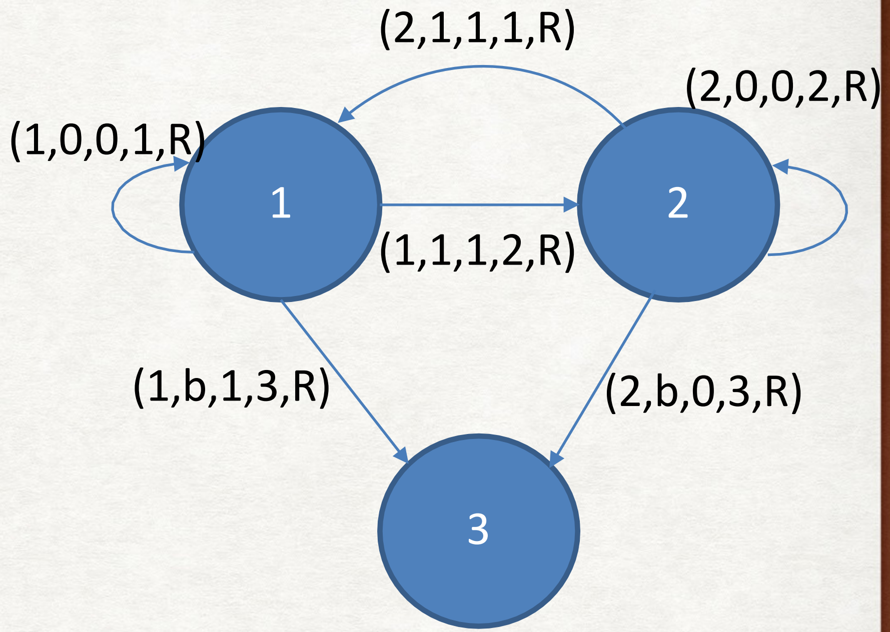
Rules:
(1, 0, 0, 1, R)
(1, 1, 1, 2, R)

(2, 0, 0, 2, R)
(2, 1, 1, 1, R)

(1, b, 1, 3, R)
(2, b, 0, 3, R)



PARITY BIT MACHINE



(1, 0, 0, 1, R)
(1, 1, 1, 2, R)
(2, 0, 0, 2, R)
(2, 1, 1, 1, R)
(1, b, 1, 3, R)
(2, b, 0, 3, R)

TURING MACHINE EXAMPLES : UNARY INCREMENTING

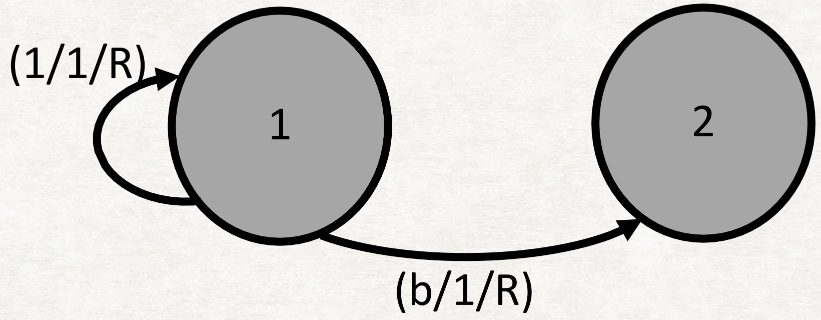
Given a unary representation of a number n,

- Change the tape to represent n+1

Unary representation: uses one symbol

- 0 = 1
- 1 = 11
- 2 = 111
- 3 = 1111

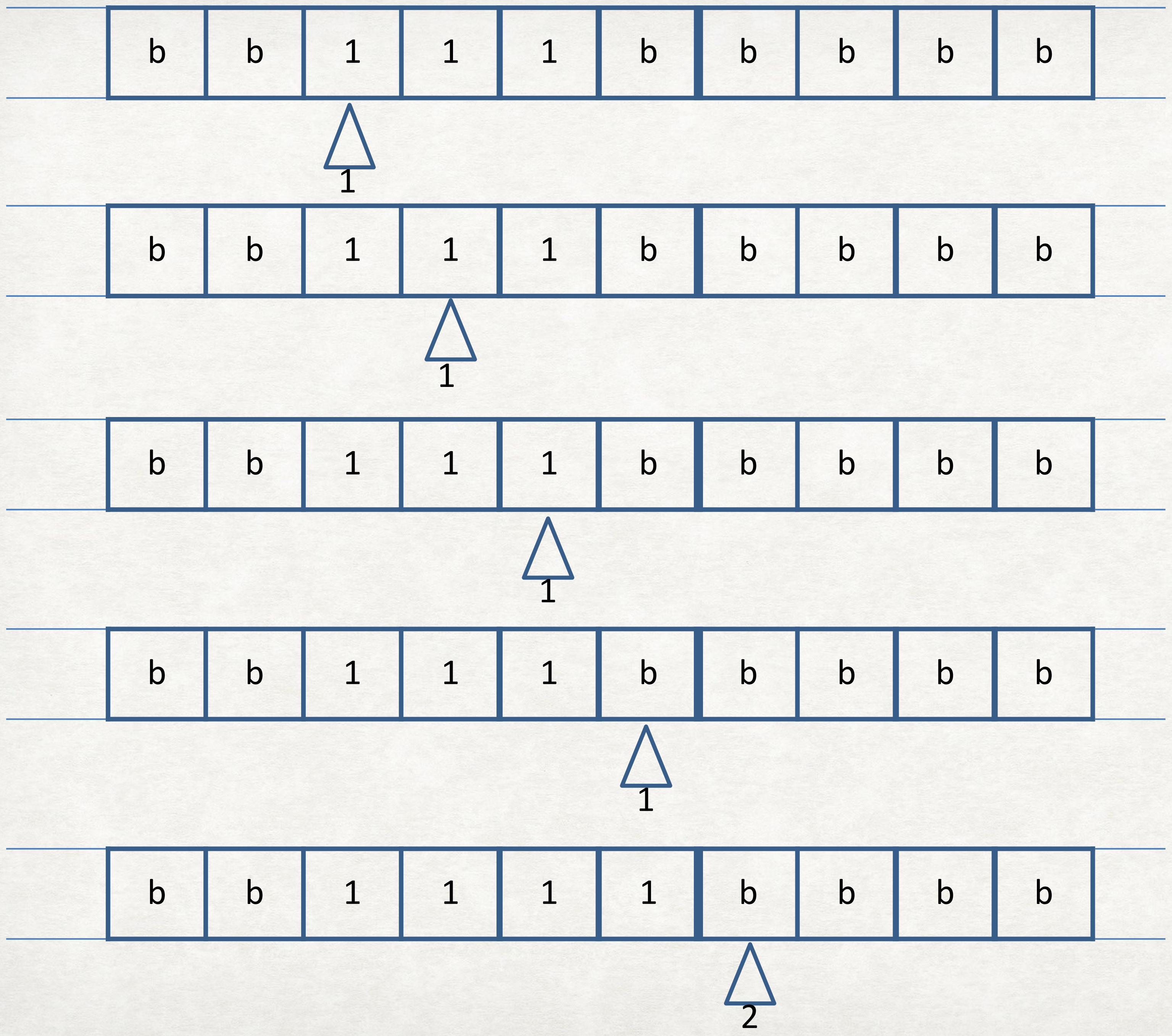
Algorithm: move to the right end and add a 1

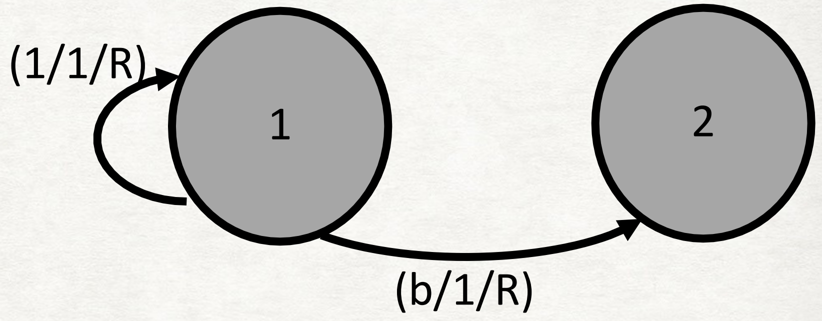


Rules:

(1, 1, 1, 1, R)
(1, b, 1, 2, R)

UNARY INCREMENTER





Rules:

(1, 1, 1, 1, R)
(1, b, 1, 2, R)

TURING MACHINE EXAMPLES

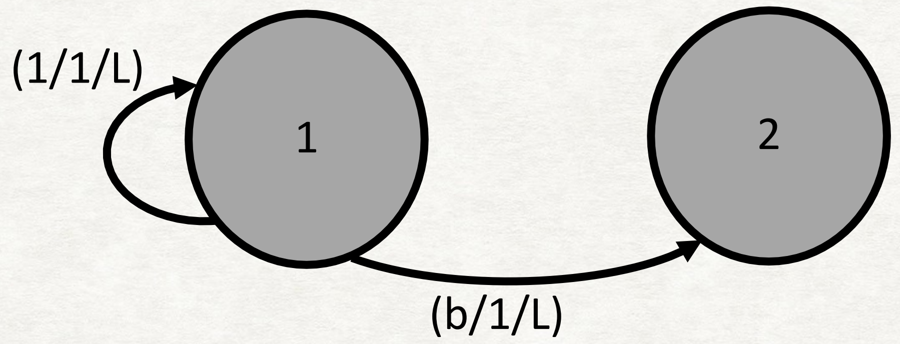
Alternative algorithm

Move left to a blank and add a 1 there.

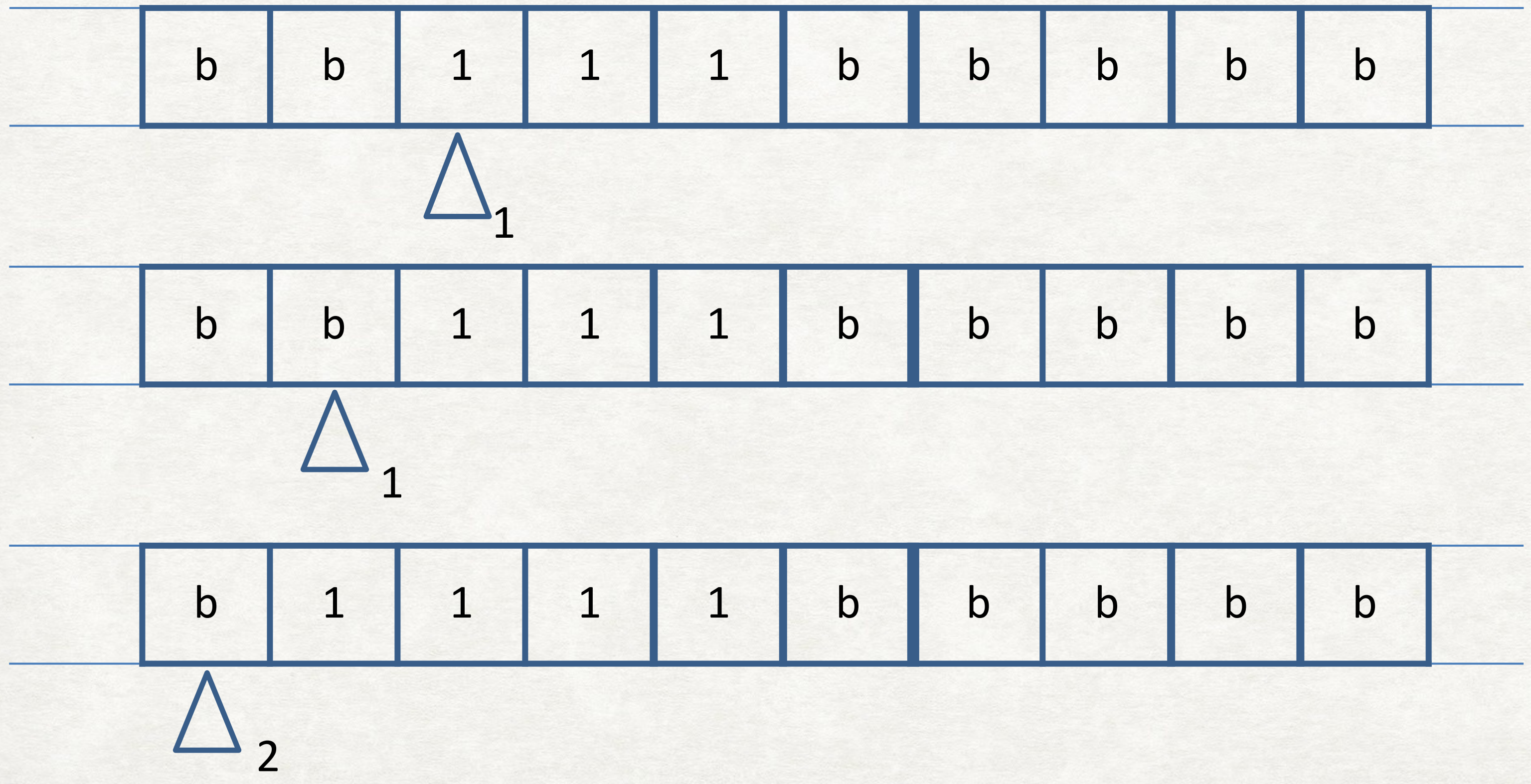
Rules:

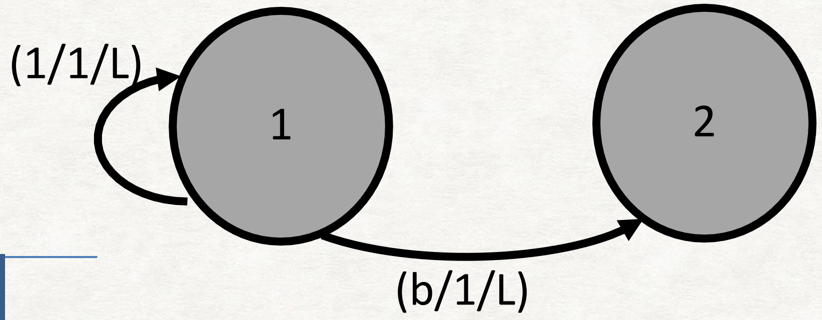
(1, 1, 1, 1, L)

(1, b, 1, 2, L)



UNARY INCREMENTOR : ALTERNATE ALGORITHM



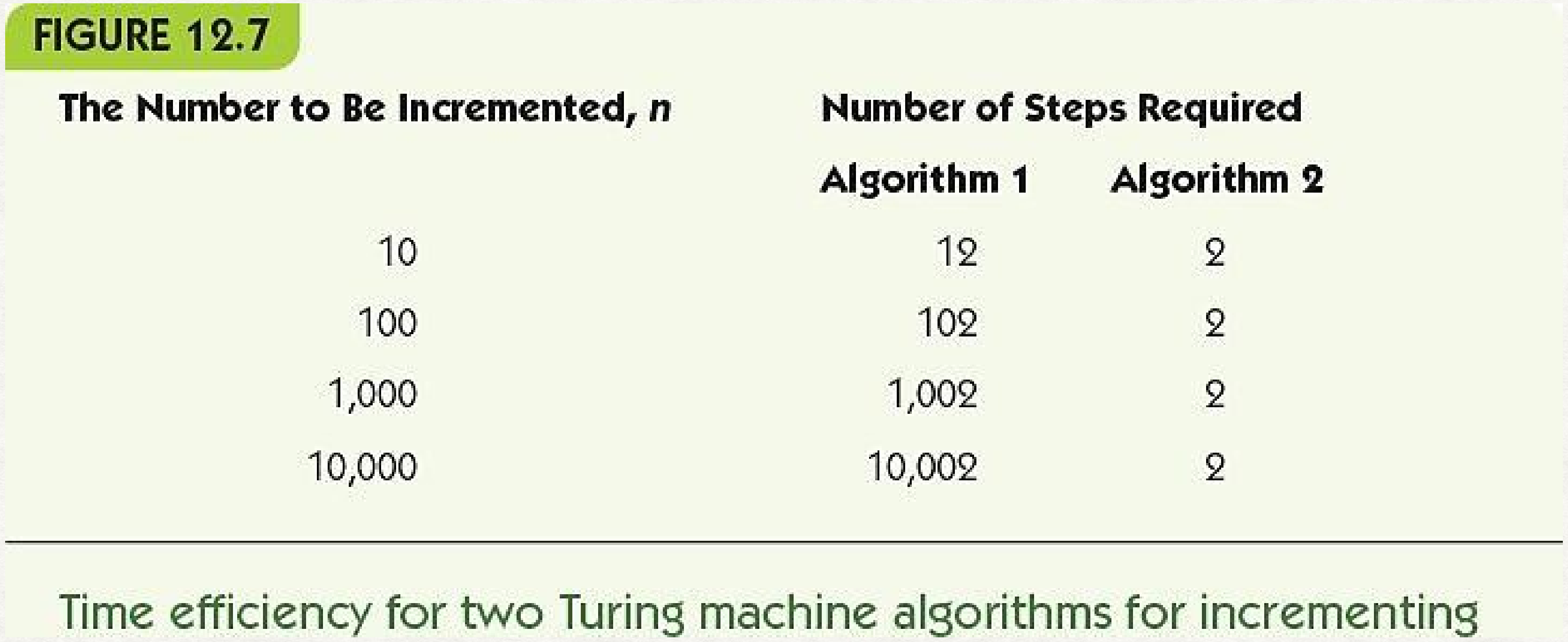


Rules:
(1, 1, 1, 1, L)
(1, b, 1, 2, L)

TURING MACHINE EXAMPLES

Algorithm 1 efficiency: number to be incremented plus 2 → n+2 steps.
Algorithm 2 efficiency: exactly 2 steps → constant steps.

FIGURE 12.7



The Number to Be Incremented, *n*	Number of Steps Required	
	Algorithm 1	Algorithm 2
10	12	2
100	102	2
1,000	1,002	2
10,000	10,002	2

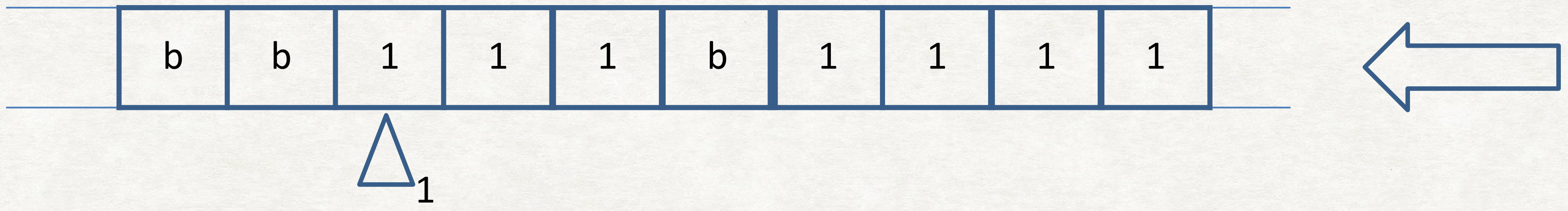
Time efficiency for two Turing machine algorithms for incrementing

TURING MACHINE EXAMPLES

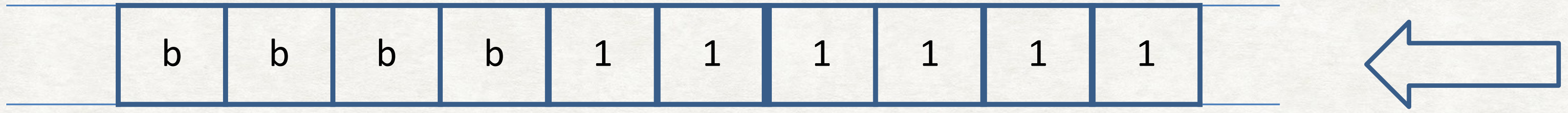
Unary addition

Given a **unary** representation of two numbers n and m, separated by one blank on the tape,

Change the tape to represent n+m

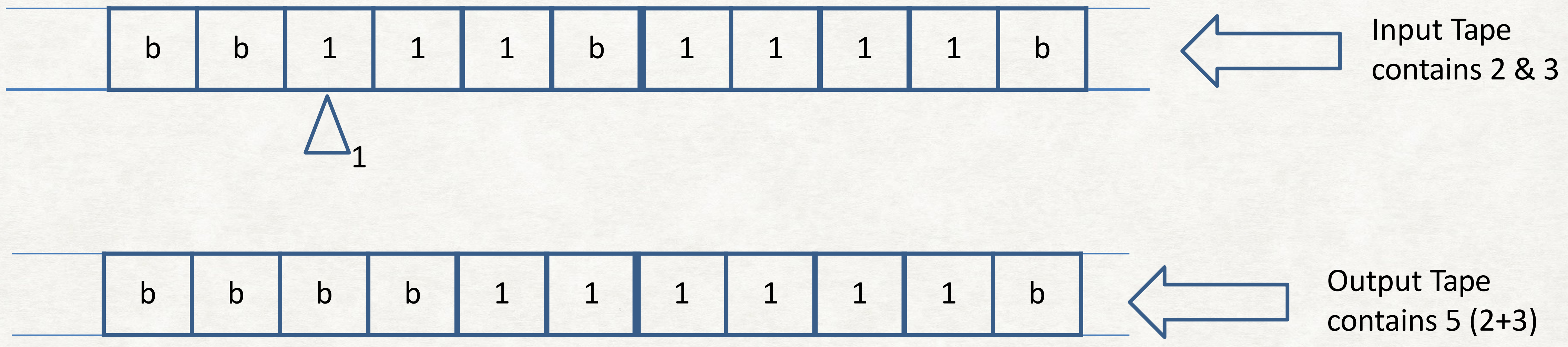


Input Tape contains 2 & 3



Output Tape contains 5 (2+3)

Turing Machine Examples



Algorithm idea

Erase leftmost 1

Erase second-to-left 1

Find a blank in the middle

Change it to a 1

Algorithm idea

Erase leftmost 1

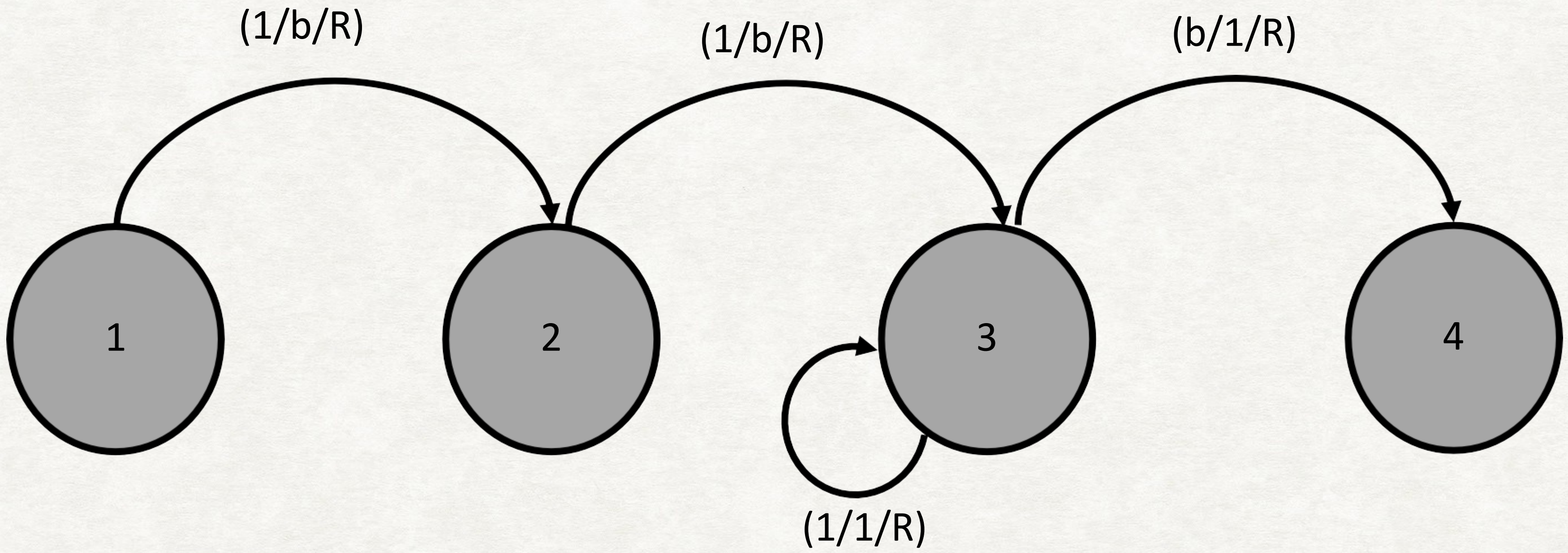
Erase second-to-left 1

Find blank in the middle

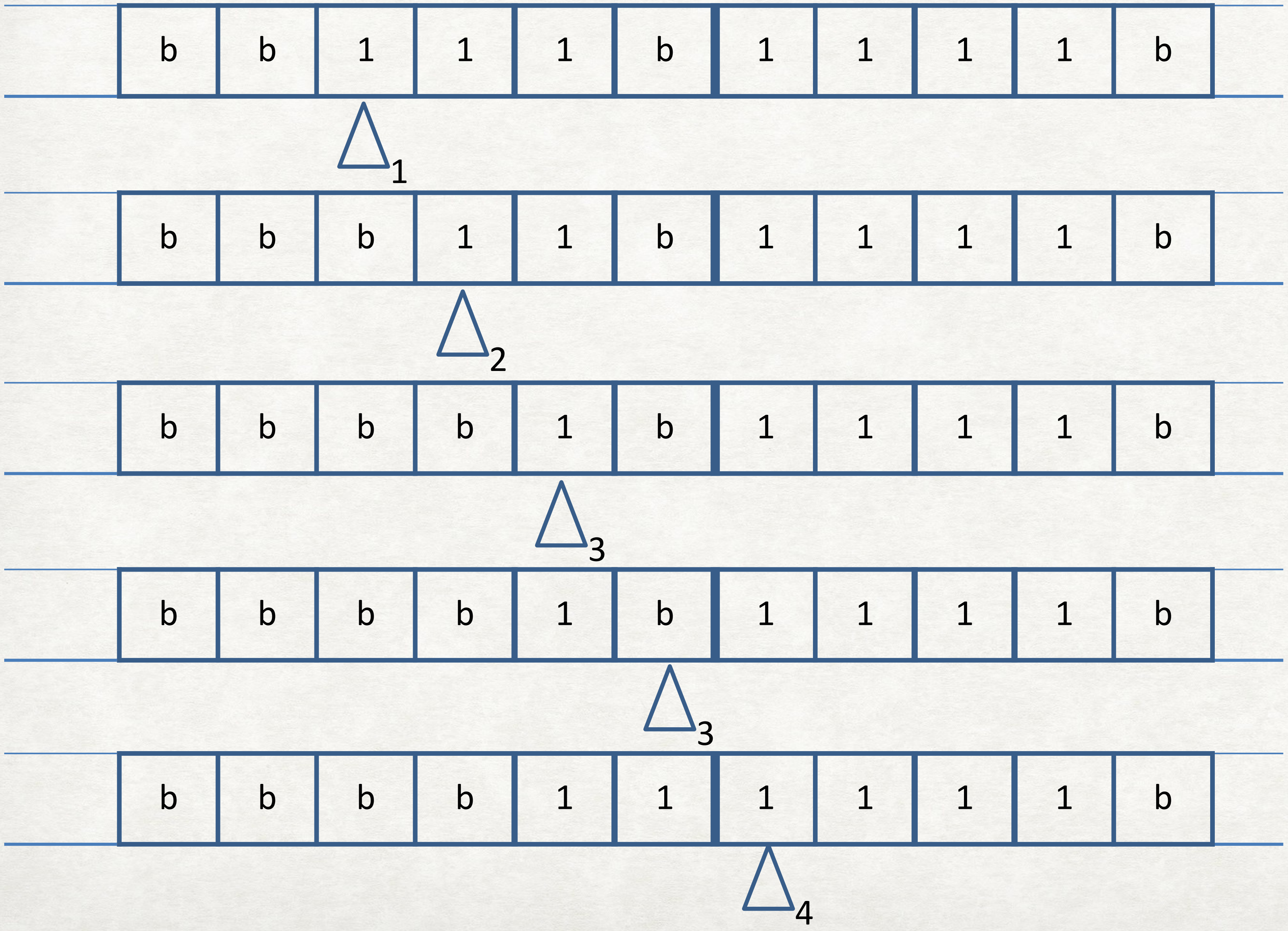
Change it to a 1

Rules:

(1, 1, b, 2, R)
(2, 1, b, 3, R)
(3, 1, 1, 3, R)
(3, b, 1, 4, R)



Unary addition - example



Rules:

(1, 1, b, 2, R)
(2, 1, b, 3, R)
(3, 1, 1, 3, R)
(3, b, 1, 4, R)

$$\begin{tikzcd}
1 \arrow[r, bend left, "(1/b/R)"] & 2 \arrow[r, bend left, "(1/b/R)"] & 3 \arrow[r, bend left, "(b/1/R)"] \arrow[loop below, "(1/1/R)"] & 4
\end{tikzcd}$$

THE CHURCH–TURING THESIS

For tasks where input and output may be represented symbolically, are there problems Turing machines can't solve that algorithms can, or vice versa?

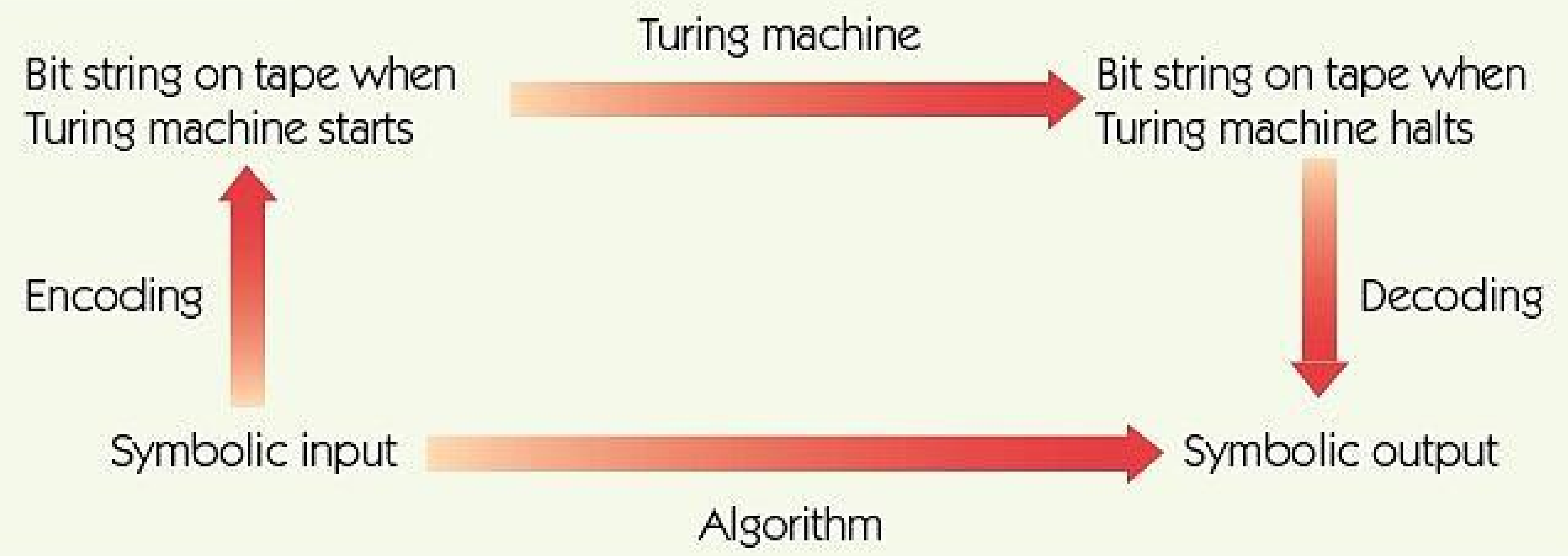
Answer: Church–Turing Thesis

> If there exists an algorithm to do a symbol manipulation task, then there exists a Turing machine to do that task.

https://www.youtube.com/watch?v=PLVCscCY4xI&t=565s&ab_channel=UpandAtom 12:55 – 15:18

The Church–Turing Thesis

FIGURE 12.9


Turing machine
Bit string on tape when Turing machine starts
Bit string on tape when Turing machine halts
Encoding
Decoding
Symbolic input
Symbolic output
Algorithm

Emulating an algorithm by a Turing machine

The Church–Turing Thesis

- "Thesis": Not proven, perhaps not provable
  - Whenever computer scientists put forward algorithmic solutions, they also tried to find Turing machines for those tasks.
    - They were always successful!
  - A number of mathematicians tried to find other models of computing agents and algorithms.
    - All of these proved equivalent to the Turing machine!

- Turing machines define the limits of what can be computed: **computability.**
  - If a problem cannot be solved by a Turing machine, then it cannot be solved algorithmically.
    - It is **uncomputable** or **unsolvable**

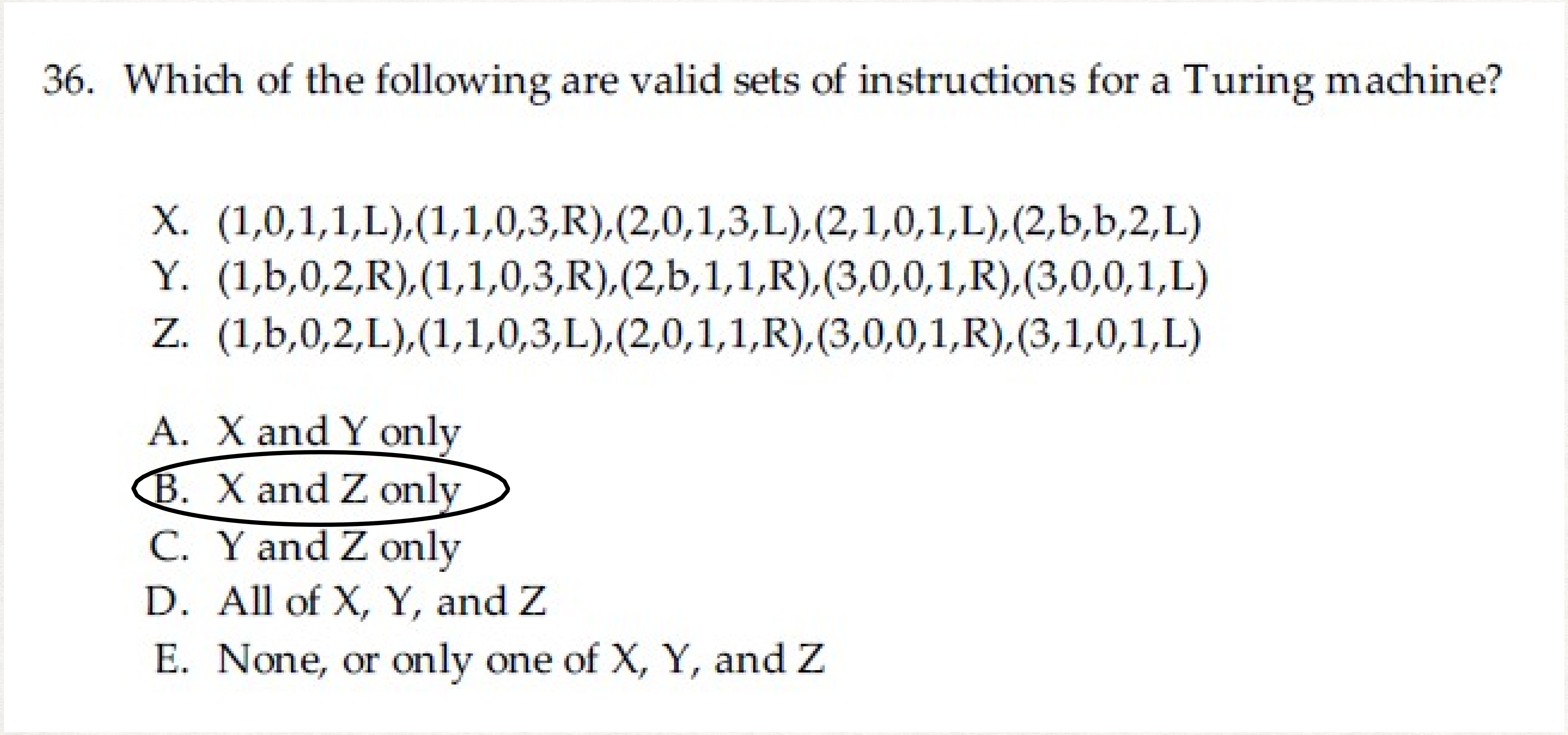
Summary

- Models allow us to predict behavior and serve as a testbed.
- A model of a computing agent must take input, have memory and state, and produce output.
- Turing machines have:
  - An infinite tape for input and memory.
  - A finite number of states.
  - Rules for changing states based on state and input, write on the tape, and move left or right.
- Turing machines are a computing agent model.

Summary

- Turing machine programs are the sets of instructions of a machine.
- Turing machine programs model algorithms.
- Examples of Turing machine programs include bit invert, parity bit, unary increment, unary addition.
- The Church–Turing thesis says that any algorithm can be performed by a Turing machine.

Sample Questions from the Exams

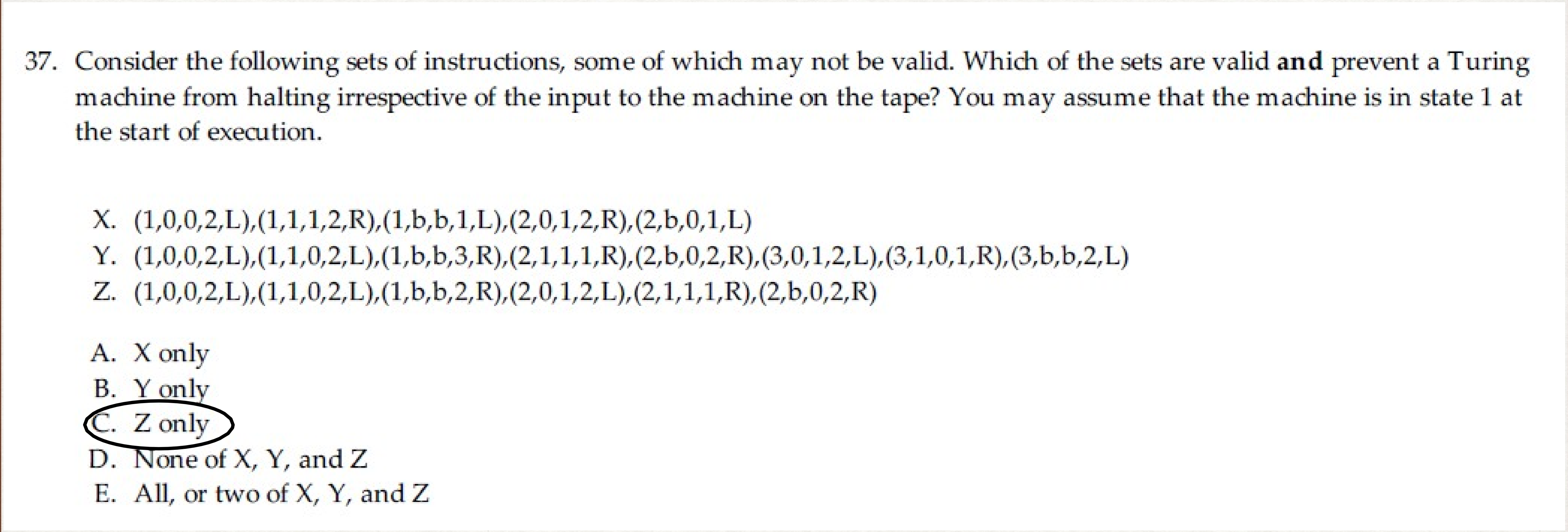


36. Which of the following are valid sets of instructions for a Turing machine?

X. (1,0,1,1,L),(1,1,0,3,R),(2,0,1,3,L),(2,1,0,1,L),(2,b,b,2,L)
Y. (1,b,0,2,R),(1,1,0,3,R),(2,b,1,1,R),(3,0,0,1,R),(3,0,0,1,L)
Z. (1,b,0,2,L),(1,1,0,3,L),(2,0,1,1,R),(3,0,0,1,R),(3,1,0,1,L)

A. X and Y only
[✓ B. X and Z only]
C. Y and Z only
D. All of X, Y, and Z
E. None, or only one of X, Y, and Z

Sample Questions from the Exams

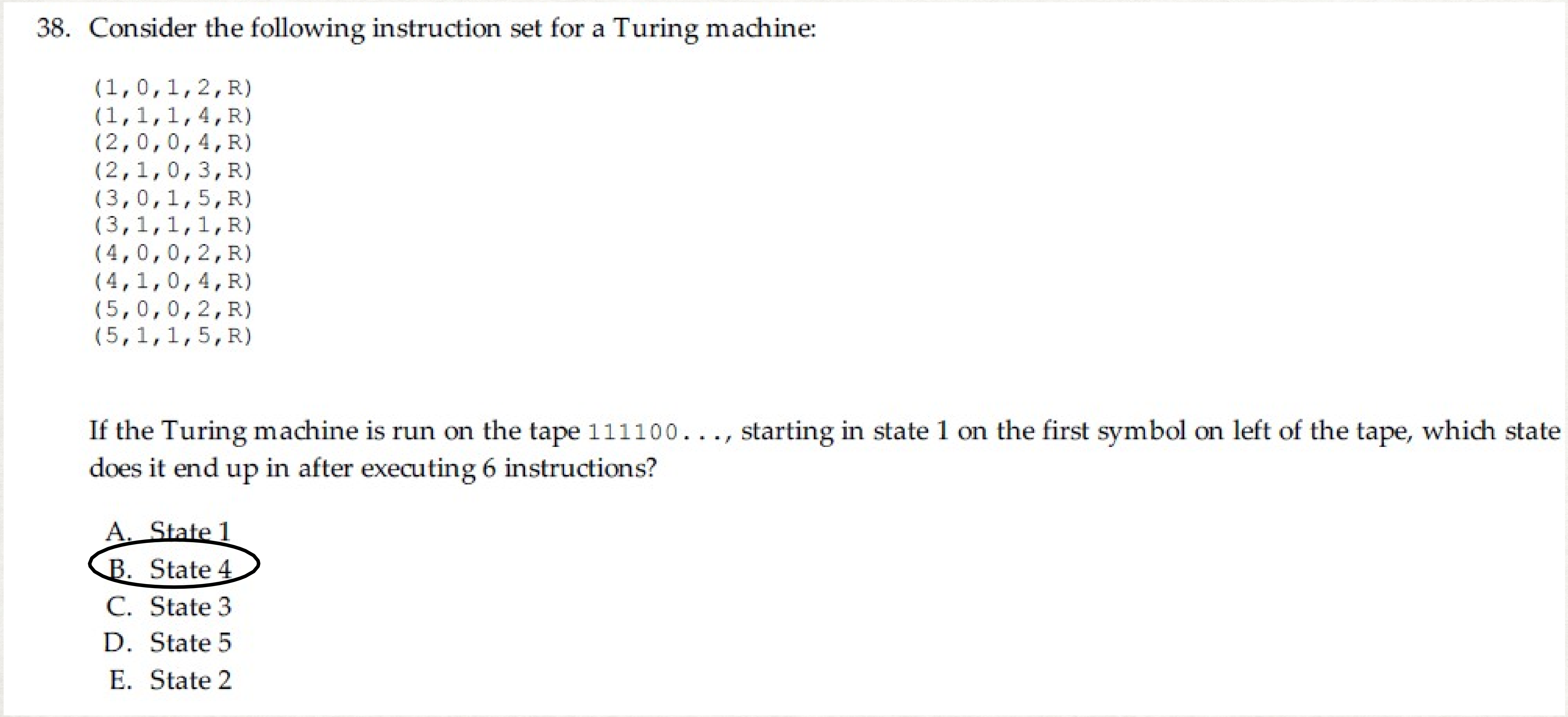


37. Consider the following sets of instructions, some of which may not be valid. Which of the sets are valid **and** prevent a Turing machine from halting irrespective of the input to the machine on the tape? You may assume that the machine is in state 1 at the start of execution.

X. (1,0,0,2,L),(1,1,1,2,R),(1,b,b,1,L),(2,0,1,2,R),(2,b,0,1,L)
Y. (1,0,0,2,L),(1,1,0,2,L),(1,b,b,3,R),(2,1,1,1,R),(2,b,0,2,R),(3,0,1,2,L),(3,1,0,1,R),(3,b,b,2,L)
Z. (1,0,0,2,L),(1,1,0,2,L),(1,b,b,2,R),(2,0,1,2,L),(2,1,1,1,R),(2,b,0,2,R)

A. X only
B. Y only
[✓ C. Z only]
D. None of X, Y, and Z
E. All, or two of X, Y, and Z

Sample Questions from the Exams



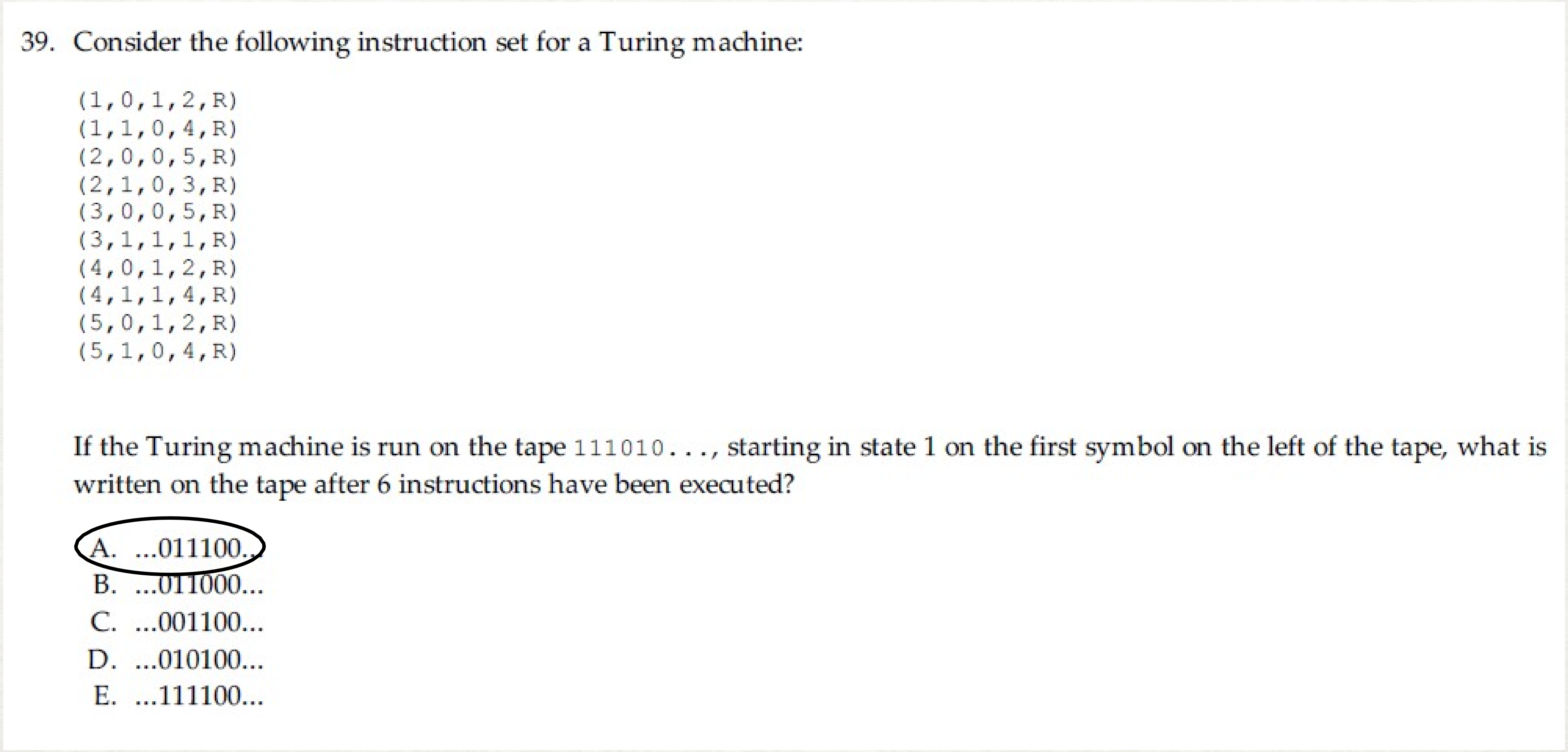
38. Consider the following instruction set for a Turing machine:

```
(1,0,1,2,R)
(1,1,1,4,R)
(2,0,0,4,R)
(2,1,0,3,R)
(3,0,1,5,R)
(3,1,1,1,R)
(4,0,0,2,R)
(4,1,0,4,R)
(5,0,0,2,R)
(5,1,1,5,R)
```

If the Turing machine is run on the tape `111100...`, starting in state 1 on the first symbol on left of the tape, which state does it end up in after executing 6 instructions?

A. State 1
[✓ B. State 4]
C. State 3
D. State 5
E. State 2

Sample Questions from the Exams



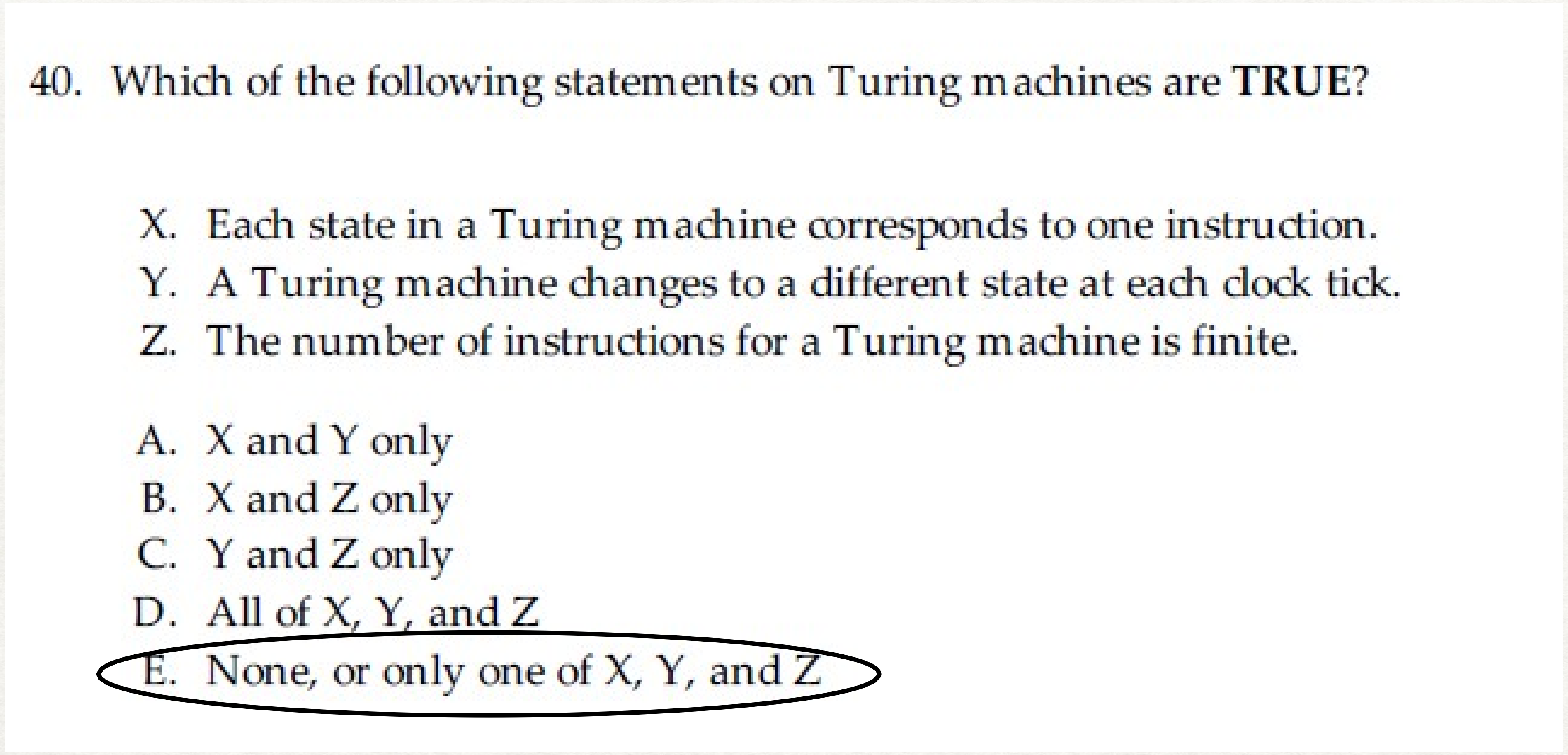
39. Consider the following instruction set for a Turing machine:

```
(1,0,1,2,R)
(1,1,0,4,R)
(2,0,0,5,R)
(2,1,0,3,R)
(3,0,0,5,R)
(3,1,1,1,R)
(4,0,1,2,R)
(4,1,1,4,R)
(5,0,1,2,R)
(5,1,0,4,R)
```

If the Turing machine is run on the tape `111010...`, starting in state 1 on the first symbol on the left of the tape, what is written on the tape after 6 instructions have been executed?

[✓ A. ...011100...]
B. ...011000...
C. ...001100...
D. ...010100...
E. ...111100...

Sample Questions from the Exams



40. Which of the following statements on Turing machines are **TRUE**?

X. Each state in a Turing machine corresponds to one instruction.
Y. A Turing machine changes to a different state at each clock tick.
Z. The number of instructions for a Turing machine is finite.

A. X and Y only
B. X and Z only
C. Y and Z only
D. All of X, Y, and Z
[✓ E. None, or only one of X, Y, and Z]